# Real-Time E-commerce Order Processing System Using Kafka

Submitted By : Siddharth Kadam (202318015)

**Objective :** Develop a Kafka-based system that manages e-commerce orders in real-time. This system will handle two critical aspects of e-commerce operations: inventory management and delivery processing.

**Following are the steps to obtain our objective :**

**Step 1 : Set Up Kafka Environment**

- **Install Kafka :** Install Kafka and configure the Below updates

1. Download Kafka from the official page (kafka_2.13-3.7.0.tgz)
2. Extract the Kafka folder
3. In config > server.properties :
   - Change the `log.dirs` to the path where you want to store the logs
4. In zookeeper.properties:
   - dataDir=D:/tmp/zookeeper
5. Start the Zookeeper server
   - `.\bin\windows\zookeeper-server-start.bat .\config\zookeeper.properties`
6. Start the Kafka server
   - `.\bin\windows\kafka-server-start.bat .\config\server.properties`
7. Create a topic
   - `.\bin\windows\kafka-topics.bat --create --topic topicBDPdemo --bootstrap-server localhost:9092`
8. Lets put some message in the kafka topic using the console "producer"
   - `.\bin\windows\kafka-console-producer.bat --topic topicBDPdemo --bootstrap-server localhost:9092`
9. Lets read the message from the kafka topic using the console "consumer"
   - `.\bin\windows\kafka-console-consumer.bat --topic topicBDPdemo --bootstrap-server localhost:9092 --from-beginning`

- **Create Kafka Topics :** Create the kafka topics for sending the messages to producer , named " inventory_order" and "delivery_order"

**Step 2 : Kafka Producers Implementations**

- **Producer for Inventory Orders (inventory_orders_producer):**
  - This producer should filter messages whose type field is inventory.
  - Implement a Kafka producer that reads inventory-related events from a data source and publishes messages with the type inventory to the inventory_order topic.
- **Producer for Delivery Orders Producer (delivery_orders_producer):**
  - This producer should filter messages whose type field is delivery.
  - Create a Kafka producer that receives delivery-related events and sends delivery-type messages to the delivery_orders topic.

**Step 3 : Kafka Consumers Implementation :**

- **Consumer for Inventory Orders (inventory_orders_consumer):**
    - Set up a Kafka consumer that subscribes to the inventory_orders topic.
    - Create logic to process inventory messages and update relevant databases or systems.

- **Consumer for Delivery Orders Producer (delivery_orders_consumer ):**
    - Create a Kafka consumer for the delivery_orders topic.
    - Create logic to manage delivery-related messages, including scheduling, updating status, and informing customers.

**Step 4 : Develop Message Filtering Logic :**

- **Message Filtering For Producer**
    - Create logic for each producer (inventory_orders_producer and delivery_orders_producer) to filter messages based on the type field in the incoming data source.
    - Only send messages to Kafka that match the specified kind (e.g. inventory or delivery).

**Conclusion:**

The development of a real-time e-commerce order processing system using Kafka provides a scalable and efficient solution for managing inventory and processing deliveries in real-time. By leveraging Kafka's distributed streaming capabilities, the system ensures reliability, scalability, and real-time responsiveness, thereby enhancing the overall efficiency of e-commerce operations.